**DS-670 Assignment 5 Mohamed Mohamar**

**Method: Algorithm Overview and Implementation in Zeppelin/R**

1. **Algorithm Overview**

Considering the linear regression modelthe functions: for *i = 1… d* in the expression: =,, …, are called *basis functions.* Now*,* instead of using linear regression, let’s allow the functions of the independent variables, or basis functions, to be adaptive.This basically means that we use parametric forms for the basis functions in which the parameter values are adapted during training. A very successful application of this model is *the feed-forward neural network*, also known as the multilayer perceptron, which is based on the *perceptron algorithm (PA).* PA is similar to the linear model but, with a step activation function:

Where *f (a) =*

The algorithm trains using the perceptron criterion, with a form of:

() = −, where M is the set of misclassified patterns.

We are looking for a weight vector such that

, and .

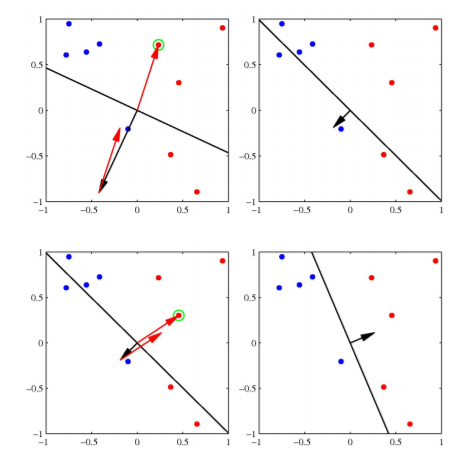
It also uses a stochastic gradient descent:

-

Now, since the perceptron function is unchanged if is multiplied by a constant, we may set = 1. We deduct the following expression:

*The perceptron convergence theorem says: If there exists an exact solution, i.e. if the training set is indeed linearly separable, then the Perceptron Algorithm will find a solution in finite number of steps.*

Following is the geometry of the perceptron algorithm:



As comparison, many applications, neural networks can be significantly more compact and faster to evaluate than a support vector machine. But this property of neural networks comes at a certain price which is the following: the likelihood function, which forms the basis for network training, is no longer a convex function of the model parameters. However, in practice, it is often worth investing substantial computational resources during the training phase in order to obtain a compact model that is fast at processing new data sets.

Linear models for regression and classification are based on linear combinations of fixed nonlinear basis functions of the form of:

*Y (****X***,) = *f*

Where *f* is a nonlinear activation function in the case of classification and is the identity in the case of regression.

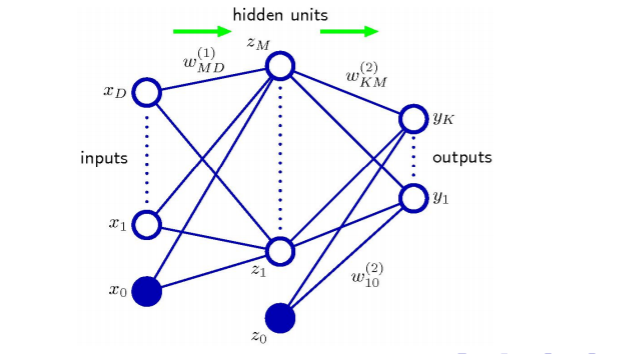
We extend this model by making the basis functions depend on parameters and then allow these parameters to be adjusted, along with the coefficients {}, during training. Hence, the basic neural network model can be described a series of functional transformations.

First we construct *M* linear combinations of the input variables … in the form of: = + where *j = 1 . . . M* and (1) indicates that the corresponding parameters are in the first layer of the network. The parameters are referred to as *weights*, the parameters as *biases*, and the quantities are known as *activations*. Each of them is then transformed using a differentiable, nonlinear activation function *h* to give *= h* ().

These quantities correspond to the outputs of the basis functions and in the context of neural networks are called *hidden* units. The nonlinear functions *h* are generally chosen to be sigmoidal functions such as the logistic sigmoid or the arc tan function. These values are again linearly combined to give output *unit activations:* = + where *k = 1 . . . K* and *K* is the total number of outputs.

This transformation corresponds to the second (2) layer of the network, and again the entities are bias parameters.

Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs noted as.



A common choice for an activation function is the following:

= σ ( ), where σ (*a*) =

Finally, we can combine all these various stages to give the overall network functions that, for sigmoidal output unit activation functions, takes the form of:

) = σ

(The *Source for the above neural network algorithm description is: DS-640: Neural Networks and Support Vector Machines - Robert Finn – Dec. 1, 2016)*

My idea is to use linear regression and neural network time series analysis to build a model to predict and forecast stock market volatility. The model will be built from the work I did last semester with Professor Robert Finn in the predictive analytics and expert design course, using R packages “LM” and “NEURALNET”.

My goal is ultimately to build a high performant stock market analysis compact model that can be used as an investment tool.

1. **Implementation in R/Zeppelin Notebook.**

For implementation, I will use the R LM and NEURALNET packages. The NEURALNET package is based on the back-propagation gradient descent algorithm. I am utilizing a double-hidden layer neural network for training.

Here is a brief summary of my data set and a description of my proceedings:

The original full data set that I am using was provided by Professor Robert Finn last semester during our DS-640 course. It is a stock market data set. It contains data about Income Statement, Cash Flow Statement, Balance Sheet, and Metrics and Ratios. Here is a list of the all the indicators or factors, as columns, included in the original data set:

**Income Statement**

**Indicator Code Name**

REVENUE Revenues

REVENUEUSD Revenues (USD)

COR Cost of Revenue

GP Gross Profit

RND Research and Development Expense

SGNA Selling, General and Administrative Expense

OPEX Operating Expenses

OPINC Operating Income

EBIT Earnings before Interest & Taxes (EBIT)

EBITUSD Earnings before Interest & Taxes (USD)

INTEXP Interest Expense

TAXEXP Income Tax Expense

CONSOLINC Consolidated Income

NETINCNCI Net Income to Non-Controlling Interests

NETINC Net Income

PREFDIVIS Preferred Dividends Income Statement Impact

NETINCCMN Net Income Common Stock

NETINCCMNUSD Net Income Common Stock (USD)

NETINCDIS Net Income from Discontinued Operations

EPS Earnings per Basic Share

EPSUSD Earnings per Basic Share (USD)

EPSDIL Earnings per Diluted Share

SHARESWA Weighted Average Shares

SHARESWADIL Weighted Average Shares Diluted

DPS Dividends per Basic Common Share

**Cash Flow Statement**

**Indicator Code Name**

NCFO Net Cash Flow from Operations

DEPAMOR Depreciation, Amortization & Accretion

SBCOMP Share Based Compensation

NCFI Net Cash Flow from Investing

CAPEX Capital Expenditure

NCFBUS Net Cash Flow - Business Acquisitions and Disposals

NCFINV Net Cash Flow - Investment Acquisitions and Disposals

NCFF Net Cash Flow from Financing

NCFDEBT Issuance (Repayment) of Debt Securities

NCFCOMMON Issuance (Purchase) of Equity Shares

NCFDIV Payment of Dividends & Other Cash Distributions

NCFX Effect of Exchange Rate Changes on Cash

NCF Net Cash Flow / Change in Cash & Cash Equivalents

**Balance Sheet**

**Indicator Code Name**

ASSETS Total Assets

ASSETSC Current Assets

ASSETSNC Assets Non-Current

CASHNEQ Cash and Equivalents

CASHNEQUSD Cash and Equivalents (USD)

RECEIVABLES Trade and Non-Trade Receivables

INTANGIBLES Goodwill and Intangible Assets

INVENTORY Inventory

LIABILITIES Total Liabilities

LIABILITIESC Current Liabilities

LIABILITIESNC Liabilities Non-Current

DEBT Total Debt

DEBTUSD Total Debt (USD)

DEBTC Debt Current

DEBTNC Debt Non-Current

DEFERREDREV Deferred Revenue

DEPOSITS Deposit Liabilities

INVESTMENTS Investments

INVESTMENTSC Investments Current

INVESTMENTSNC Investments Non-Current

PAYABLES Trade and Non-Trade Payables

PPNENET Property, Plant & Equipment Net

TAXASSETS Tax Assets

TAXLIABILITIES Tax Liabilities

EQUITY Shareholders Equity

EQUITYUSD Shareholders Equity (USD)

RETEARN Accumulated Retained Earnings (Deficit)

ACCOCI Accumulated Other Comprehensive Income

**Metrics & Ratios**

**Indicator Code Name Available Dimensions (old API)**

ASSETTURNOVER Asset Turnover ART, MRT

ASSETSAVG Average Assets ART, MRT

BVPS Book Value per Share ARQ, MRQ, ARY, MRY

CURRENTRATIO Current Ratio ARQ, MRQ, ARY, MRY

DE Debt to Equity Ratio ARQ, MRQ, ARY, MRY

DIVYIELD Dividend Yield

EBITDA Earnings before Interest, Taxes & Depreciation Amortization ARY, ARQ, ART, MRY, MRQ, MRT

EBITDAUSD Earnings before Interest, Taxes & Depreciation Amortization (USD) ARY, ARQ, ART, MRY, MRQ, MRT

EBITDAMARGIN EBITDA Margin ART, MRT

EBT Earnings before Tax

EQUITYAVG Average Equity ART, MRT

EV Enterprise Value

EVEBIT Enterprise Value over EBIT ART, MRT

EVEBITDA Enterprise Value over EBITDA ART, MRT

FCF Free Cash Flow ARY, ARQ, ART, MRY, MRQ, MRT

FCFPS Free Cash Flow per Share ARY, ARQ, ART, MRY, MRQ, MRT

FXUSD Foreign Currency to USD Exchange Rate

GROSSMARGIN Gross Margin ART, MRT

INVCAP Invested Capital ARQ, MRQ, ARY, MRY

INVCAPAVG Invested Capital Average ART, MRT

MARKETCAP Market Capitalization

NETMARGIN Profit Margin ART, MRT

PE Price Earnings Damodaran Method ART, MRT

PE1 Price to Earnings Ratio ART, MRT

PS1 Price to Sales Ratio ART, MRT

PS Price Sales Damodaran Method ART, MRT

PB Price to Book Value ARQ, MRQ, ARY, MRY

ROIC Return on Invested Capital ART, MRT

SPS Sales per Share ART, MRT

PAYOUTRATIO Payout Ratio ART, MRT

ROA Return on Average Assets ART, MRT

ROE Return on Average Equity ART, MRT

ROS Return on Sales ART, MRT

TANGIBLES Tangible Asset Value ARQ, MRQ, ARY, MRY

TBVPS Tangible Asset Book Value per Share ARQ, MRQ, ARY, MRY

WORKINGCAPITAL Working Capital

EVENT Material Corporate Events

PRICE Share Price (Adjusted Close)

SHAREFACTOR Share Factor

SHARESBAS Shares (Basic)

I am only using a subset of the data corresponding to the ARQ (As Reported Quarterly) listings, meaning where Dimension = ARQ. The Rows are the quarterly calendar date Time Series. The Columns are twenty chosen indicators along with the calculated Returns and Log of returns.

My list of twenty chosen indicators or factors is the following:

1. CURRENTRATIO Current Ratio ARQ, MRQ, ARY, MRY
2. DE Debt to Equity Ratio ARQ, MRQ, ARY, MRY
3. DIVYIELD Dividend Yield
4. FCFPS Free Cash Flow per Share ARY,MRQ, ART, MRT,

MRQ, MRY

1. GROSSMARGIN Gross Margin ART, MRT
2. PAYOUTRATIO Payout Ratio ART, MRT
3. SPS Sales per Share ART, MRT
4. NETMARGIN Profit Margin ART, MRT
5. BVPS Book Value per Share ARQ, MRQ, ARY, MRY
6. EVEBITDA Enterprise Value over EBITDA ART, MRT
7. PE1 Price to Earnings Ratio ART, MRT
8. PS1 Price to Sales Ratio ART, MRT
9. TBVPS Tangible Asset Book Value per Share ARQ, MRQ,

ARY, MRY

1. PRICE Share Price (Adjusted Close)
2. PS Price Sales Damodaran Method ART, MRT
3. EBITDAMARGIN EBITDA Margin ART, MRT
4. PB Price to Book Value ARQ, MRQ, ARY, MRY
5. REVENUE Revenues
6. EPS Earnings per Basic Share
7. PE Price Earnings Damodaran Method ART, MRT

I first used an R for loop to compute the returns from price using the formula: Returns = P(i)/P(i-1). I added it to the data set as separate column. After selecting the twenty indicators along with two columns: ticker and calendar date, I then used another R for loop to compute the log of returns and added it to the data set as a separate column. Because there are some zeros in the Returns column, using the log of returns rendered some R infinite (Inf) values in the Logreturns column. I converted those infinite values into “NAs”, before cleaning the data set from all the “NAs”.

I finally normalized all the factors using the following formula:

Normalized factor = (factor – mean (factor))/sd (factor), where mean is the mean of the column and sd is the standard deviation of the column.

I created a new R data frame consisting of all the twenty normalized factors, along with the calendar date column, the ticker column, the returns column and the log of returns column.

I found out the number of date in the data. And using an R for loop, I extracted all the data corresponding to a particular calendar date and created a data frame for each calendar date.

I will write R code to generate time series for the 20 betas derived from my linear regression of log-returns on your 20 chosen factors over the first three quarters of the dates for which I have data. I will then used these time series to forecast the 20 betas for the first date not included in the model which will me an expected return for each stock in my model over the next time period. This expected return will then be employed to rank my stocks and split them into 5 groups with group 1 having the highest expected return and group 5 the lowest expected returns. The average actual returns for each group will then be calculated for the first date not included in the model. I will then continue this process on a rolling basis for the rest of the dates for which I have data. I will perform the same task using a neural network instead of a linear model. Following is what comes next:

1. I will use time series analysis to predict neural network weights as in the case of a linear model.
2. I will incorporate factors other than fundamental factors into the model, including technical indicators, macro-economic factors, and differenced or first and second derivative information.
3. I will increase of the frequency of the model from quarterly to daily.
4. I will streamline the R code employed in the DS-640 projects, i.e. recode the project in R so that it becomes modular and compact as possible.
5. I will perform a detailed analysis for the whole market as well as at least three economic sectors.

This analysis will address not only the factors driving performance of the model but also the feasibility of investment goals of my model. This entails simulation of a portfolio controlling for risk and liquidity of the assets traded,